

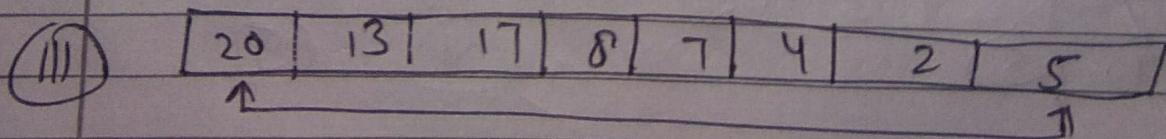
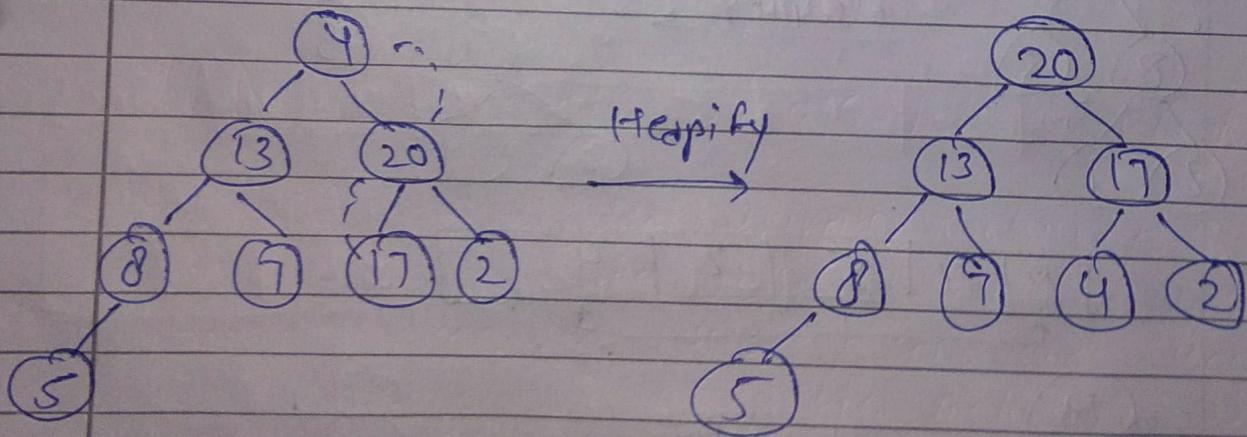
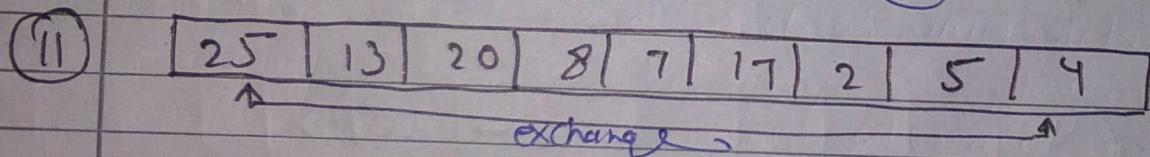
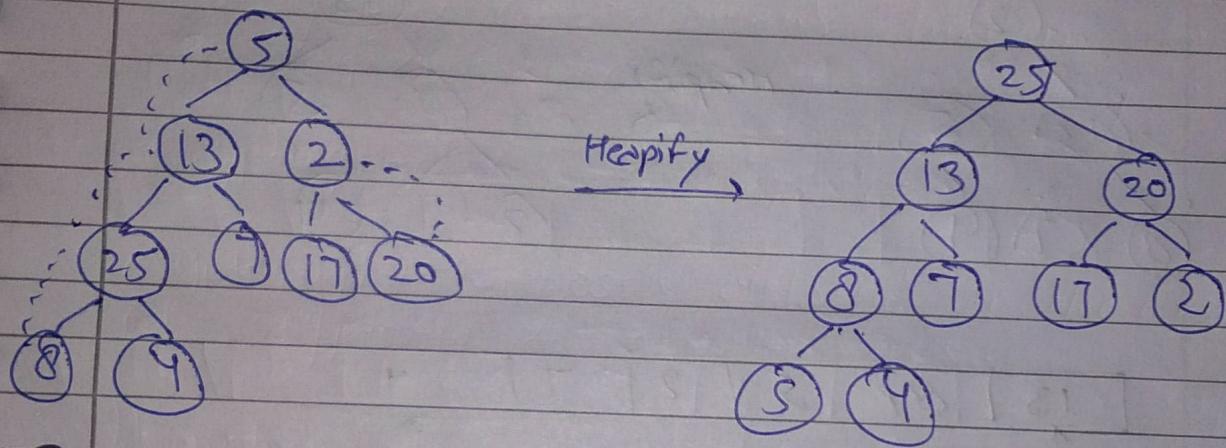
Shubham Patkar
DS [KCA 205]

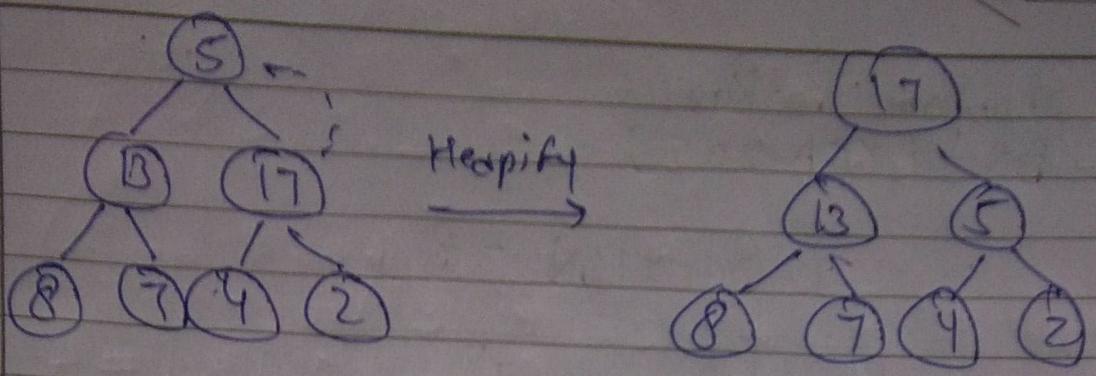
[SECTION - B]

[Answer -]

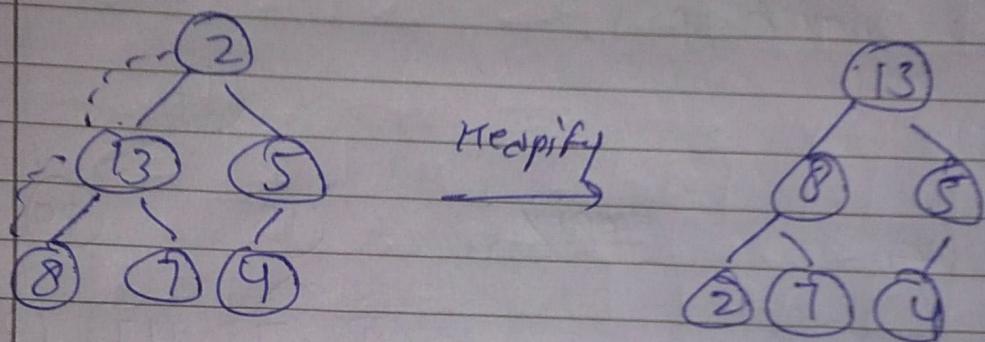
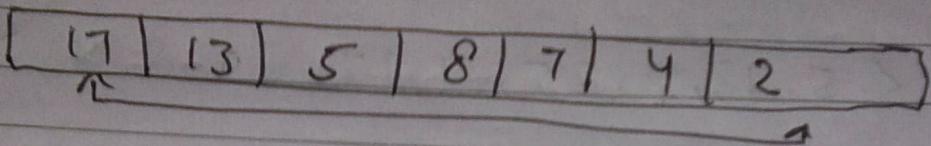
$$A = \langle 5, 13, 2, 25, 7, 17, 20, 8, 4 \rangle$$

(I) Build max heap

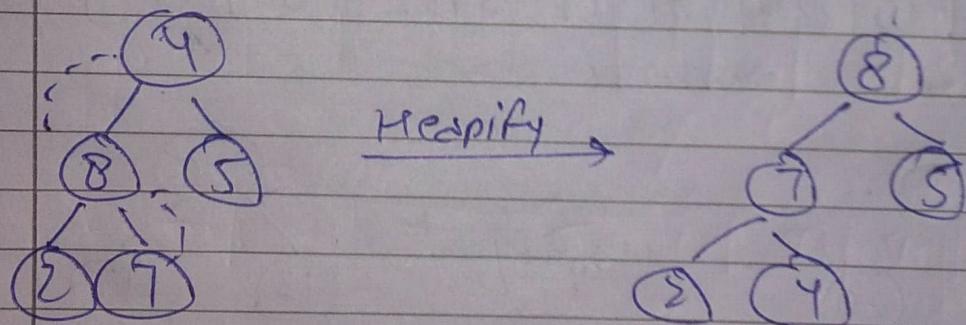
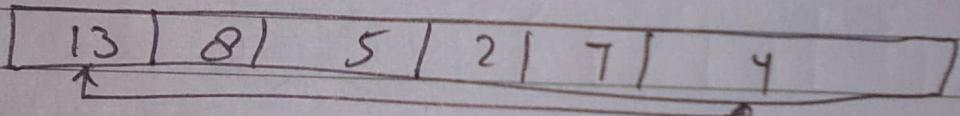




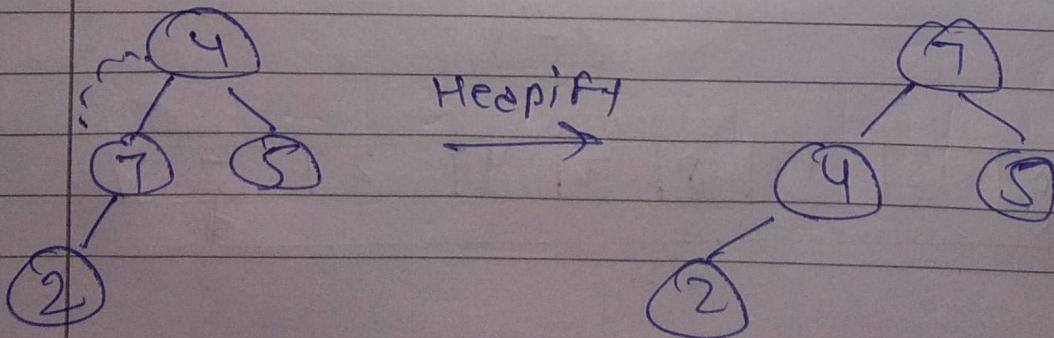
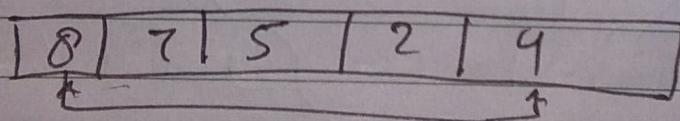
(IV)



(V)

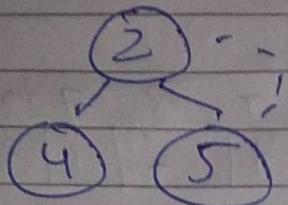
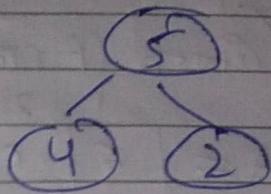


(VI)



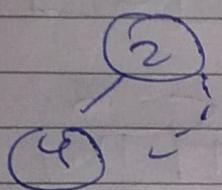
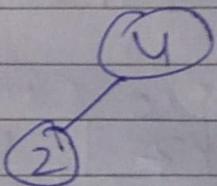
(iii)

7	4	5	2
---	---	---	---

Heapify

(iv)

5	4	2
---	---	---

Heapify

(v)

4	2
---	---

Thus sorted element of A

$$A = [2 \mid 4 \mid 5 \mid 7 \mid 8 \mid 13 \mid 17 \mid 20 \mid 25]$$

Page No. _____
Date _____

[Answer - 7]

$$A = \{6, 0, 2, 0, 1, 3, 4, 6, 1, 3, 2\} >$$

Ist - ?

Given Array

1	2	3	4	5	6	7	8	9	10	11
6	0	2	0	1	3	4	6	1	3	2

$$K = 6 \text{ [largest num]}$$

0	1	2	3	4	5	6
2	2	2	2	1	0	2

0	1	2	3	4	5	6
2	4	6	8	9	9	11

1	2	3	4	5	6	7	8	9	10	11
				2						

0	1	2	3	4	5	6
2	4	5	8	9	9	11

1	2	3	4	5	6	7	8	9	10	4
				2		13				

0	1	2	3	4	5	6
2	4	5	7	9	9	11

1	2	3	4	5	6	7	8	9	10	4
		11	2		3					

0	1	2	3	4	5	6
2	3	5	7	9	9	11

(M)

$$B = \boxed{0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \mid 10 \mid 11}$$

$$C = \boxed{\frac{0}{2} \mid \frac{1}{3} \mid \frac{2}{5} \mid \frac{3}{7} \mid \frac{4}{9} \mid \frac{5}{9} \mid \frac{6}{10}}$$

(T)

$$B = \boxed{1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \mid 10 \mid 11}$$

$$C = \boxed{\frac{0}{2} \mid \frac{1}{3} \mid \frac{2}{5} \mid \frac{3}{7} \mid \frac{4}{8} \mid \frac{5}{9} \mid \frac{6}{10}}$$

(VII)

$$B = \boxed{1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \mid 10 \mid 11}$$

$$C = \boxed{\frac{0}{2} \mid \frac{1}{3} \mid \frac{2}{5} \mid \frac{3}{6} \mid \frac{4}{8} \mid \frac{5}{9} \mid \frac{6}{10}}$$

(VIII)

$$B = \boxed{1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \mid 10 \mid 11}$$

$$C = \boxed{\frac{0}{2} \mid \frac{1}{2} \mid \frac{2}{3} \mid \frac{3}{6} \mid \frac{4}{8} \mid \frac{5}{9} \mid \frac{6}{10}}$$

(VII)

$$B = \boxed{1 \mid 0 \mid 1 \mid 1 \mid 1 \mid 2 \mid 3 \mid 3 \mid 3 \mid 1 \mid 6}$$

$$C = \boxed{\frac{0}{1} \mid \frac{1}{2} \mid \frac{2}{5} \mid \frac{3}{6} \mid \frac{4}{8} \mid \frac{5}{9} \mid \frac{6}{10}}$$

(IX)

$$B = \boxed{1 \mid 0 \mid 1 \mid 1 \mid 2 \mid 2 \mid 3 \mid 3 \mid 3 \mid 4 \mid 6}$$

$$C = \boxed{\frac{0}{1} \mid \frac{1}{2} \mid \frac{2}{4} \mid \frac{3}{6} \mid \frac{4}{8} \mid \frac{5}{9} \mid \frac{6}{10}}$$

X	$B = \boxed{0 0 1 1 2 2 3 3 4 6 }$
--------------	--

X	$C = \boxed{0 1 2 3 4 6 8 9 10 }$
--------------	---

X	$S = \boxed{0 0 1 1 2 2 3 3 4 6 }$
--------------	--

X	$C = \boxed{0 1 2 3 4 6 8 9 10 }$
--------------	---

Sorted element

X	$\boxed{0 0 1 1 2 2 3 3 4 6 }$
--------------	--

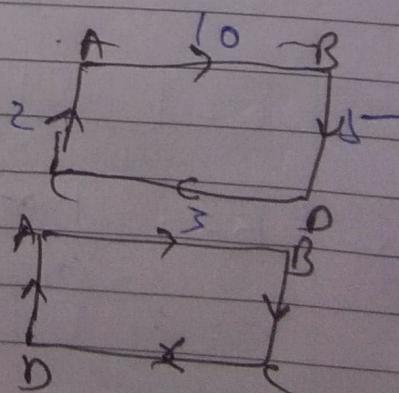
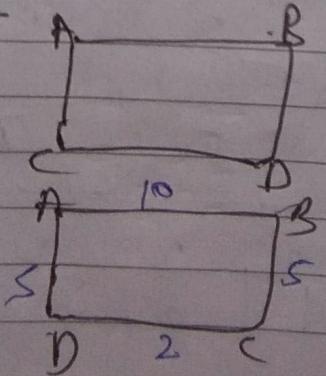
[Answer - 9]

Graph - A graph is a finite set of edges & vertices .

- It is represented by $G = (V, E)$

where , $V \rightarrow$ finite set of vertices
 $E \rightarrow$ " " " Edges .

Example -



Representation of a graph in memory

Following way of representation -

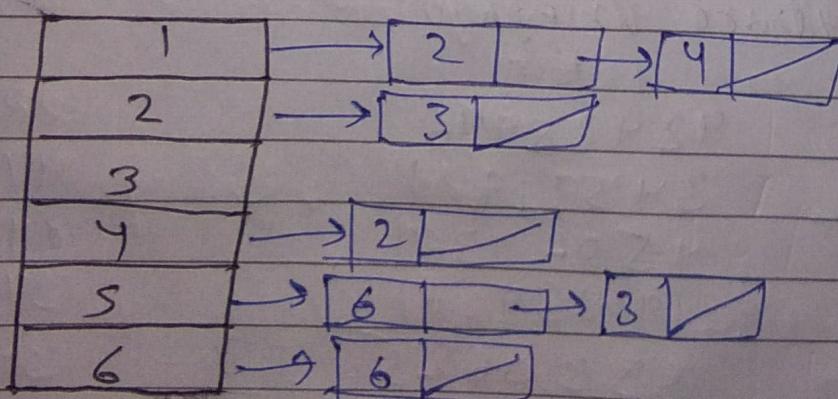
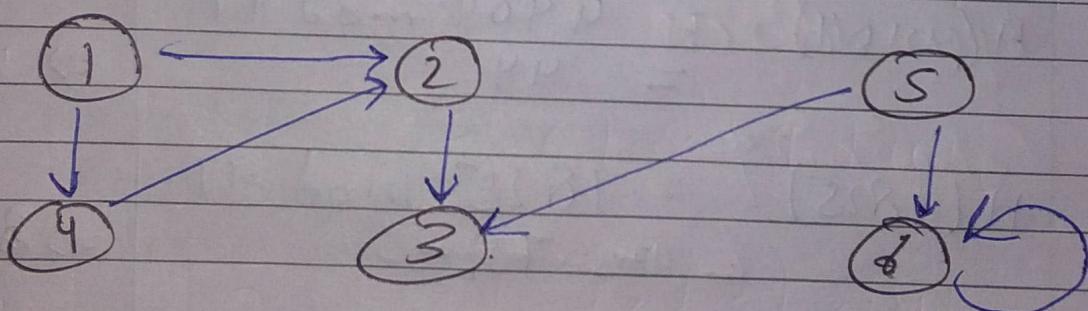
i) Sequential representation. It have ^{2 famous} techniques to represent a graph sequentially in memory

→ Adjacency matrix

→ Incidence "

ii) Linked-list representation.

If it is a graph we store all vertices in to array & each vertex is linked list.



[Answer - II]

9614, 5882, 6713, 4409, 1825

$$m = 97$$

(a) The division method

$$H(K) = K \bmod m$$

$$K \bmod m+1$$

i)

$$H(9614) = 9614 \bmod 97 \\ = 11$$

$$\Rightarrow 12$$

ii)

$$H(5882) = 5882 \bmod 97 \\ = 62$$

$$\Rightarrow 63$$

iii)

$$H(6713) = 6713 \bmod 97 \\ = 20$$

$$\Rightarrow 21$$

iv)

$$H(4409) = 4409 \bmod 97 \\ = 44$$

$$\Rightarrow 45$$

v)

$$H(1825) = 1825 \bmod 97 \\ = 79$$

$$\Rightarrow 80$$

(b) mid Square method

K

K^2

9614

92428996

28

5882

34597924

97

6713

45064369

84

4409

19439281

59

1825

3330625

50

(c) The folding method

$$H(9614) = 96 + 14 \Rightarrow 10$$

$$H(5882) = 58 + 82 \Rightarrow 40$$

$$H(6713) = 67 + 13 \Rightarrow 80$$

$$H(4409) = 44 + 09 \Rightarrow 53$$

$$H(1825) = 18 + 25 \Rightarrow 43$$

(d) folding method with using reversing

$$H(9614) = 96 + 41 \Rightarrow 37$$

$$H(5882) = 58 + 28 \Rightarrow 86$$

$$H(6713) = 67 + 31 \Rightarrow 98$$

$$H(4409) = 44 + 90 \Rightarrow 34$$

$$H(1825) = 18 + 52 \Rightarrow 70$$

[Section - A]

[Answer - 1]

Collision - A situation of collision occurs when more than one keys map to the same location.

In this situation 2 or more data element to be mapped to the same location in hash table.

Collision resolution technique,

A technique which resolve the collision is called collision resolution

It have 2 techniques -

open address

chaining

[Answer - 3]

The runtime complexity of quick sort depends on pivot.

①

Best case \rightarrow Big O N log n

②

Average case \rightarrow Big O N log n

③

Worst case \rightarrow Big O n²

[Answer - 2]

1 binary Search.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i, f, l, m, s, a[10];
    clrscr(); for(i=0; i<10; i++)
    {
        printf("Enter the num : ");
        scanf("%d", &a[i]);
    }
    printf("Enter the value for search : ");
    scanf("%d" + s);
    f = 0;
    l = a[i] - 1;
    m = (f + l) / 2;
    while (f <= l)
    {
        if (a[m] < s)
            f = m + 1;
        else if (a[m] == s)
            printf("%d Find at location %d in %s", s, m);
        else
            l = m - 1;
        m = (f + l) / 2;
    }
    if (f > l)
        printf("Not Found " + s);
    getch();
}
```

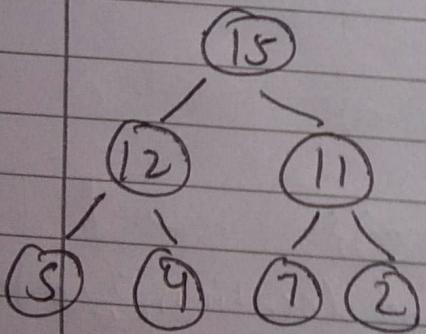
[Answer - 4]

Heap - heap is a special tree based data structure that satisfy the following heap properties.

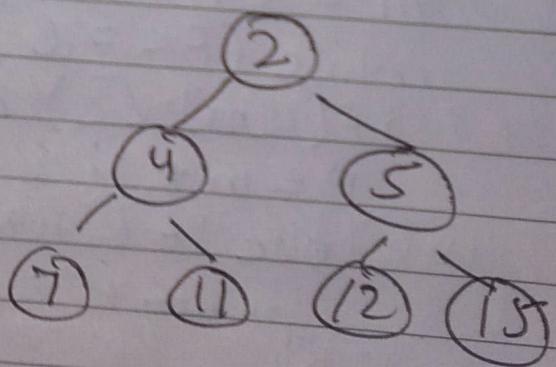
i) **shape Property** - In this, heap should be complete binary tree which means all level of the tree are full filled.

ii) **Ordered Property** - In this, a heap should be either max or min heap.

maxheap.



min heap



[Ans - 5]

```

// Selection sort [Best case complexity O(n2)]

#include <stdio.h>
#include <conio.h>

void main()
{
    int i, j, t, A[10];
    clrscr();
    for (i = 0; i < 10; i++)
    {
        printf("Enter the num = ");
        scanf(" %d", &A[i]);
    }
    for (i = 0; i < 10; i++)
    {
        for (j = i + 1; j < 10; j++)
        {
            if (A[i] > A[j])
            {
                t = A[i];
                A[i] = A[j];
                A[j] = t;
            }
        }
    }
    printf("\n Sorted element : ");
    for (i = 0; i < 10; i++)
    {
        printf("%d", A[i]);
    }
    getch();
}

```